TRANSFORM-ALIGNED LEARNED FEATURES FOR LOSSY POINT CLOUD ATTRIBUTE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Transform-based methods provide an effective framework for point cloud attribute compression by representing attributes as transform coefficients. Introducing learned spatial context into this framework requires mapping spatial representations to the transform domain, but this known basis change is often left for the network to learn implicitly. We propose Transform-Aligned Learned Features (TALF) by applying the attribute transform to learned spatial representations, explicitly aligning them with the coding targets. Our analysis shows that the resulting features exactly represent the first-order prediction term of a smooth nonlinear model, with a bounded Taylor remainder. We integrate TALF into a transform-based attribute codec with explicit coefficient prediction and conditional residual entropy modeling under a unified coefficient-domain rate–distortion objective, while retaining explicit quantization-step control. Extensive experiments across three benchmark datasets and multiple transform bases demonstrate that TALF improves rate–distortion performance over conventional and learned baselines.

1 INTRODUCTION

Point cloud attributes such as reflectance and color carry information needed for driving-scene perception and immersive 3D applications (Graziosi et al., 2020; Wang et al., 2025b). Practical attribute codecs benefit from two complementary capabilities: expressive probability modeling for efficient compression (Peng, 2025; Chen et al., 2025; Wang et al., 2023) and an explicit, interpretable quantization step (QS) for predictable rate–distortion control (Wei et al., 2025; Zhang et al., 2024). Traditional transform codecs provide a well-defined transform–quantization pipeline with reliable QS-based control, but rely on hand-crafted probability models with limited contextual capacity. Learned approaches can capture richer dependencies (Gao et al., 2025; Zhao et al., 2025; Nguyen & Kaup, 2023), yet their latent representations can obscure the link between quantization and attribute distortion (Guo et al., 2025; Ballé et al., 2021; Wang & Ma, 2022). Effectively integrating these complementary strengths is therefore a promising direction for attribute compression.

Existing methods combine conventional attribute transforms with learned entropy modeling, as in 3DAC (Fang et al., 2022), or prediction, as in DeepRAHT (Fu et al., 2026). However, the attribute transform defines the coding targets without itself aligning learned spatial representations. Consequently, the network is often left to learn this known basis change implicitly alongside coefficient prediction or probability modeling.

We propose Transform-Aligned Learned Features (TALF) to make this interface explicit. As illustrated in Figure 1, TALF applies the codec's attribute transform to every learned feature channel, with the same weighting and node ordering as the attribute signals. The local smoothness and spatial correlation of point-cloud attributes (Hu et al., 2022) motivate a local first-order approximation of a nonlinear predictor. Neighborhood-based linear prediction in conventional attribute codecs offers further practical support (Do et al., 2024). We show that transform-aligned features exactly represent this first-order prediction term and derive a bound on the higher-order remainder, providing a structural basis for learned coefficient modeling.

We instantiate TALF in an attribute codec with explicit coefficient prediction and conditional residual entropy modeling. We formulate the Laplace location parameter as an explicit coefficient prediction, jointly optimized through the entropy-rate and reconstruction-distortion terms. The resulting zero-centered residual is modeled conditionally. Orthonormality preserves reconstruction distortion,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

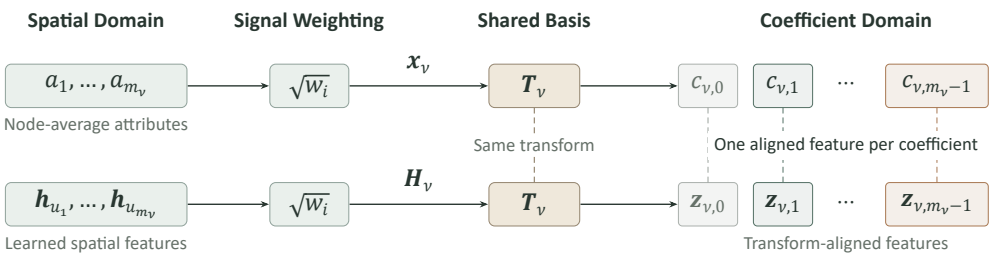


Figure 1: Transform alignment within a hierarchical block. Node-average attributes and learned spatial features undergo the same weighting and transform. Matching colors indicate coefficient–feature correspondence. Gray denotes the low-pass pair; only detail-aligned features enter the prediction and entropy-modeling heads.

enabling coefficient-domain rate–distortion learning that pairs each aligned feature with its coding target, rate, and distortion. Analytical conversion between quantization units enables one model per dataset to operate across the evaluated QS range. The alignment interface accommodates different context encoders, compatible transform bases, and entropy-coding methods, with experiments covering a normalized graph basis and RAHT. With learned entropy coding, TALF achieves BD-Rate savings of 20.93% on Ford, 14.01% on KITTI, and 9.71% on ScanNet relative to our G-PCC anchor.

Our contributions are:

- We introduce **transform-aligned learned features** through a parameter-free interface that applies the attribute transform to learned context features and establish its first-order alignment property with a bounded nonlinear remainder.
- We formulate a unified **coefficient-domain rate–distortion objective** that couples an explicit coefficient prediction with conditional entropy modeling of the resulting zero-centered residual.
- We integrate these components into a **QS-compatible lossy point cloud attribute coding framework**. Experiments on reflectance and color across datasets and transform bases demonstrate its rate–distortion benefits and the modularity of the alignment interface.

2 Related Work

Conventional Transform-Based Attribute Compression Traditional point cloud attribute compression exploits geometry–attribute correlation through prediction and transform coding. The MPEG Geometry-based Point Cloud Compression (G-PCC) standard provides prediction-, lifting-, and RAHT-based attribute coding tools (Schwarz et al., 2019; MPEG 3DG, 2023). In particular, RAHT can be viewed as a geometry-adaptive variant of the Haar wavelet transform, recursively applying occupancy-weighted two-point transforms over the point-cloud octree hierarchy (De Queiroz & Chou, 2016). Graph-transform methods instead regard attributes as signals over geometry-derived graphs (Shao et al., 2017; Xu et al., 2021). Early work established blockwise graph Fourier coding for regular and sparse point clouds (Zhang et al., 2014; Cohen et al., 2016). RA-GFT incorporates multiresolution region masses through a Q-normalized graph Laplacian (Pavez et al., 2020), while SSGT recursively constructs normalized-Laplacian subspace transforms (Chen et al., 2020). Grounded in classical signal analysis, these methods provide well-defined transform–quantization pipelines. TALF builds on this structure by using the same transform to generate attribute coefficients and align the learned features used to model them.

Learning-Based Lossy Attribute Compression Existing learned methods broadly follow two routes according to the representation being quantized and entropy-coded: learned latent representations or explicit attribute transform coefficients. The first route includes Deep-PCAC, a geometry-conditioned point-based autoencoder (Sheng et al., 2022), and Unicorn, which employs universal multiscale conditional coding for both lossy and lossless attribute compression (Wang et al., 2025a). LVAC encodes learned latent representations in the form of RAHT coefficients for attribute recon-

struction with local coordinate-based networks (Isik et al., 2022). These methods offer expressive nonlinear modeling, but their reliance on learned representations makes rate–distortion control less interpretable.

The second route learns prediction or probability models for explicit attribute transform coefficients. 3DAC retains RAHT and learns a context-adaptive probability model for arithmetic coding of the quantized transform coefficients (Fang et al., 2022). DeepRAHT predicts node attributes through interpolation and learned compensation, then applies RAHT to obtain coefficient predictions for residual coding (Fu et al., 2026). It uses conventional zero run-length coding rather than a learned conditional entropy model. However, coefficient prediction or probability modeling does not itself ensure alignment between learned spatial representations and transform coefficients. This mismatch motivates explicit feature–coefficient alignment.

3 TRANSFORM-ALIGNED LEARNED FEATURES

Our goal is to construct coefficient-specific representations for probability modeling. We establish two theoretical properties in a general transform-block setting. First, aligned features exactly represent the first-order term of a shared nonlinear signal predictor followed by the transform, with a second-order bound on the nonlinear remainder. Second, orthonormality preserves reconstruction error, enabling coefficient-domain rate–distortion learning.

3.1 TRANSFORM-BLOCK FORMULATION

Consider a local transform block $\mathcal{B}_\nu = \{1, \dots, m_\nu\}$ with scalar signal samples $\mathbf{x}_\nu = [x_{\nu,1}, \dots, x_{\nu,m_\nu}]^\top$. The block may be formed by any deterministic construction available at both codec ends. We present one signal channel; multiple channels can be processed separately or conditionally. A known analysis transform produces

$$\mathbf{c}_\nu = \mathbf{T}_\nu \mathbf{x}_\nu, \tag{1}$$

where $\mathbf{T}_\nu \in \mathbb{R}^{m_\nu \times m_\nu}$ and coefficient indices in $\mathbf{c}_\nu$ start at zero.

For the analysis below, we consider orthonormal transforms that separate a constant signal component from its detail components. These properties are standard in many classical transform-coding constructions (Goyal, 2001; Mallat, 2009). With the low-pass, or DC, basis first and the remaining local detail, or AC, rows denoted by $\mathbf{T}_{\nu,\mathrm{AC}}$, we write

$$\mathbf{T}_\nu^\top \mathbf{T}_\nu = \mathbf{I}, \qquad \mathbf{T}_{\nu,\mathrm{AC}} \mathbf{1} = \mathbf{0}. \tag{2}$$

The first row is $\mathbf{d}_\nu^\top = \mathbf{1}^\top / \sqrt{m_\nu}$, so the low-pass coefficient equals $\sqrt{m_\nu}$ times the block mean. A constant signal has zero detail coefficients. The detail basis may be any orthonormal complement of the constant direction.

We use unweighted signals here to expose the alignment principle. Appendix A.2 extends the analysis to weighted coordinates, and Section 4 specializes it to point-count-weighted hierarchical attribute coding.

3.2 TRANSFORM ALIGNMENT OF LEARNED FEATURES

For each element $i \in \mathcal{B}_\nu$, an encoder extracts $\mathbf{h}_{\nu,i} = \mathcal{E}_\theta(i, \mathcal{C}_\nu) \in \mathbb{R}^d$, where $\mathcal{C}_\nu$ denotes any context available at both codec ends. Stack these features in the same element order as the signal:

$$\mathbf{H}_\nu = [\mathbf{h}_{\nu,1}, \dots, \mathbf{h}_{\nu,m_\nu}]^\top \in \mathbb{R}^{m_\nu \times d}.$$

Rows of $\mathbf{H}_\nu$ identify block elements, whereas entries of $\mathbf{c}_\nu$ identify transform modes that generally combine multiple elements. Directly assigning element features to coefficient slots therefore leaves a known basis change implicit, which we call *feature–coefficient misalignment*.

TALF removes this mismatch by applying the same analysis transform to every feature channel:

$$\mathbf{Z}_\nu = \mathbf{T}_\nu \mathbf{H}_\nu \in \mathbb{R}^{m_\nu \times d}. \tag{3}$$

The k-th row $\mathbf{z}_{\nu,k}^\top$ is paired with coefficient $c_{\nu,k}$. Each aligned feature provides an informative representation for learned coefficient modeling, specifically coefficient prediction and residual probability estimation.

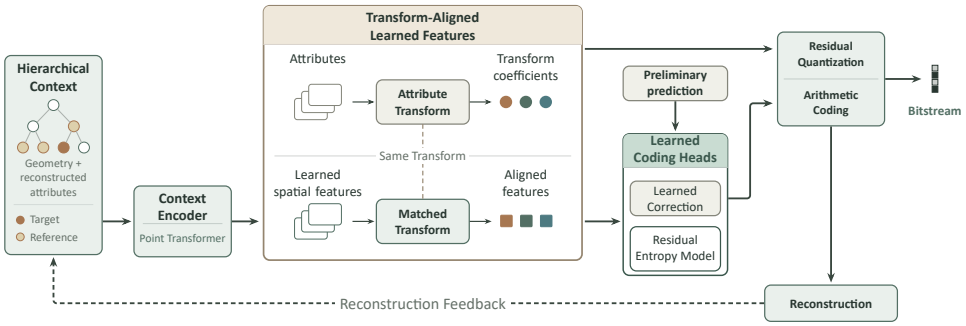


Figure 2: Overview of a TALF-based point-cloud attribute codec. Learned spatial features are aligned with attribute coefficients to guide coefficient prediction and residual entropy modeling.

3.3 Theoretical Analysis and Coefficient-Domain Learning

Local first-order alignment. We first establish why transform-aligned features provide a suitable representation for coefficient prediction. We fix a block and omit the subscript ν throughout this subsection. Assume that its signal samples admit one common nonlinear function f:

$$x_i = f(\mathbf{h}_i) + \epsilon_i,$$

where ϵ_i is the residual not explained by f. The function may vary across blocks and is assumed to be twice continuously differentiable in a neighborhood of the convex hull of the block's features. This regularity condition holds for networks with smooth activations such as GELU (Hendrycks & Gimpel, 2016). Define the feature center as $\bar{\mathbf{h}} = m^{-1}\sum_i \mathbf{h}_i$. Expanding f about $\bar{\mathbf{h}}$ and stacking the elements gives

$$\begin{aligned}\mathbf{x} &= \mathbf{H}\underbrace{\nabla f(\bar{\mathbf{h}})}_{\mathbf{g}} + \underbrace{\left[f(\bar{\mathbf{h}}) - \nabla f(\bar{\mathbf{h}})^\top \bar{\mathbf{h}}\right]}_{\xi}\mathbf{1} + \boldsymbol{\tau} + \boldsymbol{\epsilon} \\ &= \mathbf{H}\mathbf{g} + \xi\mathbf{1} + \boldsymbol{\tau} + \boldsymbol{\epsilon},\end{aligned}$$

Here $\boldsymbol{\tau}$ is the second-order Taylor remainder. Since $\mathbf{T}_{\mathrm{AC}}\mathbf{1} = \mathbf{0}$, the shared intercept contributes only to the low-pass coefficient. Applying the detail rows to this expansion therefore yields

$$\boxed{\mathbf{c}_{\mathrm{AC}} = \mathbf{Z}_{\mathrm{AC}}\mathbf{g} + \mathbf{T}_{\mathrm{AC}}(\boldsymbol{\tau} + \boldsymbol{\epsilon})}, \tag{4}$$

where $\mathbf{Z}_{\mathrm{AC}}$ contains the detail rows of $\mathbf{Z}$, and the remaining error comprises the transformed Taylor remainder and unexplained residual. For a globally shared affine predictor $f(\mathbf{h}) = \mathbf{g}^\top\mathbf{h} + \xi$, the Taylor remainder vanishes and $\mathbf{g}$ is identical across blocks. A shared linear head on the aligned features then exactly reproduces the transformed detail predictions.

For nonlinear f, if $\|\nabla^2 f(\mathbf{h})\|_2 \le \kappa$ along every segment from $\bar{\mathbf{h}}$ to an element feature, the nonlinear discrepancy obeys

$$\|\mathbf{T}_{\mathrm{AC}}\boldsymbol{\tau}\|_2 \le \frac{\kappa}{2}\left(\sum_i \|\mathbf{h}_i - \bar{\mathbf{h}}\|_2^4\right)^{1/2}. \tag{5}$$

The higher-order contribution is therefore small when the function has low local curvature and the within-block features have limited dispersion. Aligned features thus exactly represent the first-order prediction term, with a bounded higher-order remainder, providing a structural basis for learned coefficient modeling. In practice, a globally shared nonlinear head approximates these block-dependent prediction relationships. Appendix A.2 provides the complete derivation.

Coefficient-domain rate–distortion objective. We next formulate the training objective in the coefficient domain. Orthonormality preserves squared reconstruction error independently of the local prediction model. For any reconstruction $\hat{\mathbf{x}}$, with $\hat{\mathbf{c}} = \mathbf{T}\hat{\mathbf{x}}$, Parseval's identity gives

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \|\mathbf{c} - \hat{\mathbf{c}}\|_2^2. \tag{6}$$

Within each block, fixing the low-pass reconstruction makes detail-coefficient distortion equivalent to signal-domain distortion up to a constant term. We therefore optimize the following coefficient-domain rate–distortion objective:

$$\mathcal{L}_{\mathrm{RD}} = D_{\mathrm{coeff}} + \beta R = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \left[(c_k - \hat{c}_k)^2 + \beta \left(-\log_2 p(q_k \mid \mathcal{C}_k) \right) \right], \tag{7}$$

where $\mathcal{K}$ indexes the coefficients modeled by the learned components, q_k is the transmitted quantization index, $\mathcal{C}_k$ is the coding context available at both codec ends, and $\beta > 0$ controls the rate–distortion trade-off. This coefficient-wise formulation associates every aligned feature with its target coefficient, reconstruction error, and estimated rate. The alignment principle is independent of the context encoder, prediction head, entropy model, and the choice of basis within the transform setting above.

4 TALF-Based Attribute Codec

We instantiate the alignment principle of Section 3 in a hierarchical attribute codec. As shown in Figure 2, a context encoder extracts node features from hierarchical context, TALF aligns these features with the attribute coefficients, and learned coding heads predict coefficients and their residual distribution parameters. Reconstructed attributes then provide context for subsequent coding steps.

4.1 Hierarchical Context and Aligned Feature Construction

We organize the point-cloud geometry using an octree, with the occupied children of each parent forming a transform block. Each child node u_i represents w_i points, and its attribute a_i is their average attribute. With $\mathbf{a}_\nu = [a_1, \ldots, a_{m_\nu}]^\top$ and $\mathbf{M}_\nu = \mathrm{diag}(w_1, \ldots, w_{m_\nu})$, the transform input is $\mathbf{x}_\nu = \mathbf{M}_\nu^{1/2} \mathbf{a}_\nu$. Each entry $x_i = \sqrt{w_i}\, a_i$ is the DC coefficient propagated from an internal child's subtree, or the original attribute at a leaf where $w_i = 1$. This corresponds to $\mathbf{S}_\nu = \mathbf{M}_\nu^{1/2}$ in Appendix A.2.

For an occupied child node u_i in block $\mathcal{B}_\nu$, we construct a reference set $\mathcal{R}(u_i) = \mathcal{R}_{\mathrm{par}}(u_i) \cup \mathcal{R}_{\mathrm{cur}}(u_i)$. We first select nearby parent-level nodes and replace those whose child attributes have already been reconstructed with their occupied children. $\mathcal{R}_{\mathrm{par}}$ contains the remaining parent-level references, and $\mathcal{R}_{\mathrm{cur}}$ contains the replacement child nodes. Each reference contains relative coordinates, reconstructed attributes, and its own reconstructed prediction residual. Higher-level coordinates are averages of child coordinates weighted by point counts.

Our implementation uses an adaptive Point Transformer encoder (You et al., 2024). Query-masked attention combines the reference features with their relative positions while suppressing padded entries. The resulting features are aggregated and mapped to a representation $\mathbf{h}_{u_i} = \mathcal{E}_\theta(u_i, \mathcal{R}(u_i))$. To match these transform inputs, we weight each learned representation by $\sqrt{w_i}$ and stack the results in the same element order to form $\mathbf{H}_\nu$. This is the point-count-weighted specialization of $\mathbf{H}_\nu$ in Section 3.2.

As a concrete instantiation of the general transform in Section 3.1, we use a normalized graph-Laplacian basis (Chen et al., 2020) for both attribute analysis and feature alignment. The graph construction and basis conventions are specified in Appendix A.1. RAHT (De Queiroz & Chou, 2016) provides an alternative basis through the same interface. Only features corresponding to valid AC coefficients enter the learned heads; DC coefficients follow the hierarchical propagation rule.

4.2 Coefficient Prediction and Residual Entropy Modeling

Each aligned feature $\mathbf{z}_{\nu,k}$ supplies the context for two coefficient-specific quantities: a coefficient prediction and the parameters of its residual entropy model.

Explicit coefficient prediction. We use a deterministic preliminary predictor together with a learned correction. For each occupied child node u_i, we interpolate the reconstructed attributes of the three nearest reference nodes in $\mathcal{R}(u_i)$ using inverse-distance weighting. Collecting these node-average attribute predictions gives $\mathbf{a}_\nu^{\mathrm{pre}}$. Square-root weighting followed by the attribute transform

yields the preliminary coefficient prediction

$$\boldsymbol{\mu}_{\nu}^{\mathrm{pre}} = \mathbf{T}_{\nu}\mathbf{M}_{\nu}^{1/2}\mathbf{a}_{\nu}^{\mathrm{pre}}.$$

The coefficient-prediction head g_{μ} produces a learned correction from the aligned feature. Adding this correction to the preliminary prediction gives the final coefficient prediction $\mu_{\nu,k}^{\mathrm{p}}$:

$$\mu_{\nu,k}^{\mathrm{p}} = \mu_{\nu,k}^{\mathrm{pre}} + \mu_{\nu,k}^{\mathrm{corr}}, \qquad \mu_{\nu,k}^{\mathrm{corr}} = g_{\mu}(\mathbf{z}_{\nu,k}). \tag{8}$$

For channel χ with quantization step Δ_{χ}, the quantized residual index $q_{\nu,k}$ and reconstructed coefficient $\hat{c}_{\nu,k}$ are given by

$$q_{\nu,k} = \mathcal{Q}\left(\frac{c_{\nu,k} - \mu_{\nu,k}^{\mathrm{p}}}{\Delta_{\chi}}\right), \qquad \hat{c}_{\nu,k} = \mu_{\nu,k}^{\mathrm{p}} + \Delta_{\chi} \cdot q_{\nu,k}. \tag{9}$$

Here $\mathcal{Q}$ is the scalar quantizer in index units. Unlike a location parameter used solely for entropy modeling, the explicit prediction affects both the transmitted residual index and the reconstructed coefficient. Appendix A.5 further analyzes these two effects through the integer and fractional components of the learned correction.

Zero-centered residual entropy modeling. We model the residual indices with a zero-centered discretized Laplace distribution. The entropy head predicts only a positive scale parameter $b_{\nu,k}$, assigning probability and estimated rate

$$p_{\nu,k}(q) = \int_{q-\frac{1}{2}}^{q+\frac{1}{2}} \mathrm{Lap}(t; 0, b_{\nu,k})\,\mathrm{d}t, \qquad R_{\nu,k} = -\log_2 p_{\nu,k}(q_{\nu,k}).$$

After transform alignment, a shared MLP maps each $\mathbf{z}_{\nu,k}$ to a compact hidden representation, from which lightweight heads predict $\mu_{\nu,k}^{\mathrm{corr}}$ and $b_{\nu,k}$. For color, corrections are predicted jointly, while scale prediction is conditioned on previously decoded channel indices in the order $\mathrm{Cb} \to \mathrm{Cr} \to \mathrm{Y}$.

The learned components are trained with the coefficient-domain rate–distortion objective in Eq. (7). The forward pass uses hard quantization indices, with a straight-through estimator (Bengio et al., 2013) for backpropagation. We also evaluate conventional run-length coding of the residual indices, as reported in Section 5.2.

4.3 Coding Procedure and Quantization-Step Control

Coding and reconstruction proceed from coarse to fine through the octree, starting from the root DC. Combining the AC coefficients reconstructed by Eq. (9) with the DC supplied by the coarser level gives $\hat{\mathbf{c}}_{\nu}$. The node-average attributes $\hat{\mathbf{a}}_{\nu} = \mathbf{M}_{\nu}^{-1/2}\mathbf{T}_{\nu}^{\top}\hat{\mathbf{c}}_{\nu}$ then provide context for subsequent blocks and finer levels. At the leaf level, they constitute the final point cloud reconstruction.

Conversion between quantization units. Converting learned coefficient corrections and entropy scales between quantization units enables a single model to operate across QS values with explicit quantization control. Let $\Delta_{\chi}^{\mathrm{r}}$ and $\Delta_{\chi}^{\mathrm{t}}$ denote the reference training QS and target QS, respectively. In implementation, the prediction head parameterizes the correction by $\mu_{\chi}^{q,\mathrm{r}}$ in units of the reference quantization step, so that $\mu_{\chi}^{\mathrm{corr}} = \Delta_{\chi}^{\mathrm{r}} \cdot \mu_{\chi}^{q,\mathrm{r}}$. With b_{χ}^{r} denoting the predicted scale in the same units, the target-QS parameters are

$$\rho_{\chi} = \frac{\Delta_{\chi}^{\mathrm{r}}}{\Delta_{\chi}^{\mathrm{t}}}, \qquad \mu_{\chi}^{q,\mathrm{t}} = \rho_{\chi}\mu_{\chi}^{q,\mathrm{r}}, \qquad b_{\chi}^{\mathrm{t}} = \rho_{\chi}b_{\chi}^{\mathrm{r}}. \tag{10}$$

This conversion preserves the coefficient-domain correction. For cross-channel conditioning, the decoded symbol q_{χ}^{t} is rescaled to reference units before embedding as $q_{\chi}^{\mathrm{r}} = q_{\chi}^{\mathrm{t}}/\rho_{\chi}$.

5 Experiments

We now present the experimental settings, compression performance, and ablation studies.

Table 1: Comparisons of coding performance using average BD-Rate (%) against the baseline methods and runtime (s/frame). G-PCCv33 is the BD-Rate anchor, except for Unicorn†, which uses the reported Unicorn-G-PCC (RAHT21) anchor.

(a) Reflectance compression

Dataset	3DAC	Unicorn	TALF
Ford	$+1.81$	$-5.63^{\dagger}$	$\mathbf{-20.93}$
KITTI	$+9.89$	$-4.68^{\dagger}$	$\mathbf{-14.01}$

(b) Color compression

Dataset	3DAC	DeepRAHT	TALF-Neural	TALF-RL	TALF-Hybrid
ScanNet	$+3.85$	-6.65	-9.71	-12.26	$\mathbf{-15.07}$

(c) Runtime (Enc./Dec.)

Method	Ford	KITTI	ScanNet
G-PCCv33	1.5 / 0.8	0.6 / 0.3	1.8 / 1.4
3DAC	3.8 / 4.5	3.6 / 4.9	4.3 / 5.0
DeepRAHT	–	–	3.1 / 3.1
TALF	7.9 / 7.7	9.0 / 8.9	8.5 / 8.4

5.1 Experimental Setup

Datasets. We evaluate color compression on ScanNet (Dai et al., 2017), using 1,513 scans for training and 100 scans for testing. The geometry is quantized at a spatial resolution of 2 mm, preserving point uniqueness and one-to-one correspondence with the associated colors. Reflectance experiments use the KITTI LiDAR dataset (Geiger et al., 2012) and Ford (Pandey et al., 2011). KITTI sequences 00–10 are used for training and 11–21 for testing; Ford sequence 01 is used for training, with sequences 02 and 03 reserved for testing. Both reflectance datasets are voxelized at a resolution of 1 mm.

Baselines and metrics. Our default baseline, denoted G-PCCv33, is TMC13v33 (MPEG 3DG, 2026) configured with predictive RAHT. We compare with 3DAC (Fang et al., 2022), Unicorn (Wang et al., 2025a), and DeepRAHT (Fu et al., 2026). We retrain 3DAC on the same training data as TALF and evaluate it on our test sets. We report the released Unicorn reflectance results with their original Unicorn-G-PCC (RAHT21) anchor for reference. For DeepRAHT, we evaluate the executable released in its GitHub repository on our color test set without retraining.

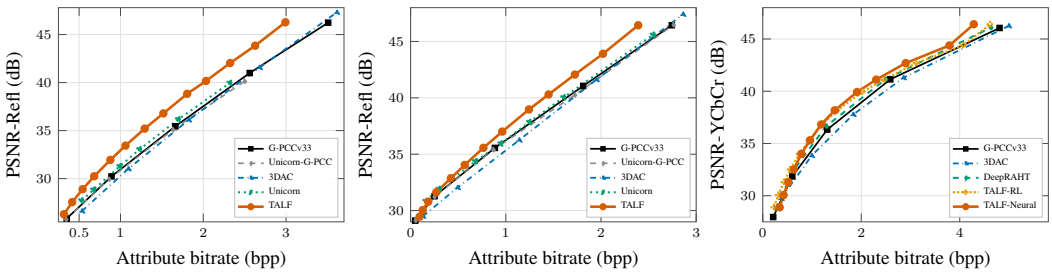
Attribute bitrate is reported in bits per input point (bpp). We report PSNR-Refl for reflectance and PSNR-YCbCr for color, with color MSE computed by weighting the Y, Cb, and Cr channel MSEs in a 6:1:1 ratio. Per-channel color PSNR is reported in the appendix. BD-Rate is computed using a Bjøntegaard fit over the common quality interval (Bjøntegaard, 2001).

Implementation. Experiments use an NVIDIA Tesla T4 GPU and an Intel Xeon Gold 6248 CPU at 2.50 GHz. For each dataset, a single model is evaluated at 13 base QS settings from 4 to 64 using the conversion in Section 4.3. Network and training configurations are given in Table 4, and the causal batching procedure is described in Appendix A.4.

5.2 Overall Rate–Distortion Performance

Figure 3 and Table 1 show consistent rate–distortion improvements over G-PCCv33 across three datasets, with bitrate savings reaching 20.93% on Ford. The gains extend from outdoor LiDAR reflectance to indoor color scans, demonstrating the effectiveness of the codec across different attribute types and scene conditions.

On ScanNet, we distinguish three variants with identical coefficient predictions and reconstructions: TALF-Neural uses the learned residual entropy model, TALF-RL uses conventional run-length coding, and TALF-Hybrid uses learned entropy coding for $\mathrm{QS} \leq 16$ and run-length coding otherwise, using a validation-selected threshold fixed for all test samples. All three variants outperform 3DAC and DeepRAHT in average BD-Rate, with TALF-Hybrid achieving the largest savings. As shown in Table 1(c), G-PCCv33 is the fastest method, while TALF has moderately higher runtimes than 3DAC and DeepRAHT.



(a) Ford. (b) KITTI. (c) ScanNet.

Figure 3: Rate–distortion curves for Ford and KITTI reflectance and ScanNet color.

Table 2: Analysis of transform alignment on Ford. (a) Residual energy reduction is relative to zero-innovation prediction. (b) Prediction MSE is evaluated on valid AC coefficients before quantization. (c) BD-Rate is relative to G-PCCv33. All reductions, increases, and BD-Rate values are in %.

(a) Assessing the first-order approximation

Predictor	Nonlinearity	Parameters	Innovation MSE	Residual energy reduction
Shared linear predictor	None	513	139.2518	30.37
Shared one-hidden-layer MLP	GELU	263,169	135.2843	32.35

(b) Feature-mapping ablation relative to TALF

Feature mapping	Prediction MSE increase	BD-Rate vs. TALF
Weighted centering	+30.45	+13.32
Learned transform	+28.11	+10.17

(c) Transform-basis comparison

Method	Transform basis	BD-Rate
TALF-RAHT	RAHT	−20.35
TALF	Graph	**−20.93**

5.3 Analysis of Transform Alignment

We examine the first-order approximation, the effectiveness of explicit feature alignment, and its applicability across transform bases on Ford.

Assessing the first-order approximation. We examine whether a single linear predictor shared across child nodes captures most of the prediction benefit provided by a nonlinear model. Specifically, we freeze the feature encoder and compare a shared linear predictor with a nonlinear MLP predictor for predicting the child-attribute innovations after the deterministic preliminary predictor. As shown in Table 2(a), after convergence, the shared linear predictor yields approximately 94% of the MLP's residual energy reduction with approximately $500\times$ fewer trainable parameters. These results suggest that linear prediction accounts for most of the observed gain, supporting the first-order analysis in Section 3.3.

Feature-mapping ablation. We replace TALF's explicit feature transform with two alternative mappings while keeping the remaining coding framework unchanged. The *weighted-centering* variant forms child representations as $\widetilde{\mathbf{h}}_{u_i} = \sqrt{w_i}(\mathbf{h}_{u_i} - \bar{\mathbf{h}}_w)$, where $\bar{\mathbf{h}}_w = \sum_j w_j \mathbf{h}_{u_j} / \sum_j w_j$. The point-count weights w_i are defined in Section 4.1. The first $m_\nu - 1$ centered features are paired with AC coefficients in coding order without applying $\mathbf{T}_\nu$. The *learned-transform* variant instead uses a shared nonlinear mapping across child features, allowing data-driven alignment without imposing the codec's transform basis.

Table 2(b) shows that weighted centering and the learned mapping both increase prediction MSE, with BD-Rate penalties of 13.32% and 10.17%, respectively, relative to TALF. These results indicate that removing the constant component alone is insufficient, while learning the mapping does not fully recover the benefit of explicit transform alignment.

Table 3: Ablation of explicit coefficient correction on Ford. BD-Rate (%) is reported relative to G-PCCv33, No μ, and Integer μ_n^q.

Prediction variant	BD-Rate vs. G-PCCv33	BD-Rate vs. No μ	BD-Rate vs. Integer μ_n^q
No μ	-7.68	0.00	–
Integer μ_n^q	-15.50	-8.46	0.00
Full μ	-20.93	-14.02	-6.00

Applicability across transform bases. To examine dependence on the transform basis, we instantiate the same alignment interface with RAHT. The corresponding RAHT operator (Taquet & Lasserre, 2020) is applied to both attribute signals and learned features, while the other coding components remain unchanged. As shown in Table 2(c), TALF-RAHT achieves 20.35% BD-Rate savings over G-PCCv33 on Ford, only 0.58 percentage points below TALF. These results support the applicability of the alignment interface across the two tested transform bases.

5.4 Explicit Coefficient Correction and Entropy Coding

Effect of explicit coefficient correction. We decompose the learned correction in quantization-index units as $\mu^q = \mu_n^q + \mu_f^q$, where $\mu_n^q = \mathrm{round}(\mu^q)$ and μ_f^q is the remaining fractional component. The No μ variant uses the deterministic preliminary predictor without learned coefficient correction; Integer μ_n^q adds the grid-aligned learned correction; Full μ also includes its fractional part. Each variant uses the same trained model and codec configuration but advances its own reconstructed causal context and produces an actual arithmetic-coded bitstream across the evaluated QS range.

Without learned coefficient correction, No μ still models residual uncertainty through conditional scales and achieves 7.68% BD-Rate savings over G-PCCv33 (Table 3). Integer correction further recenters the coded symbols. The fractional component shifts the reconstruction grid within a quantization step and enables learned adjustments to quantization decisions toward lower rate–distortion cost. Together, these effects yield an additional 6.00% BD-Rate reduction over Integer μ_n^q. Appendix A.5 gives the corresponding local symbol statistics and coefficient-error analysis.

Effect of the entropy-coding method. To examine entropy-backend dependence, we retain TALF's coefficient prediction and feature alignment while switching between neural and run-length coding (MPEG 3DG, 2020), keeping reconstructed coefficients fixed (Figure 3(c)). TALF-Neural uses fewer bits at fine QS, whereas TALF-RL is more efficient at coarse QS, where residuals are sparse. Part of this gap arises from the finite-precision entropy coder. Our 16-bit arithmetic CDF uses a 2049-symbol alphabet, capping the probability of any symbol at $31/32$. Consequently, even all-zero color residuals incur an approximate rate floor of 0.137 bpp. For ScanNet scene 0707_00 at the coarsest tested QS, the neural and run-length payloads are 0.237 and 0.112 bpp, respectively. The latter falls below this floor, showing that improved probability estimates alone cannot close the gap.

6 Conclusion

We presented Transform-Aligned Learned Features (TALF), which applies the attribute transform to align learned spatial representations with their coefficient targets. Our framework combines TALF, explicit coefficient prediction, and conditional residual entropy modeling under a unified coefficient-domain rate–distortion objective, while preserving quantization-step control. Results on reflectance and color show coding gains over conventional and learned baselines, with both the normalized graph basis and RAHT supporting the effectiveness of the alignment principle.

More broadly, this work suggests that attribute transforms can structure learned representations by making the known basis change explicit, while neural models learn contextual dependencies. The TALF alignment interface has the potential to accommodate different context encoders, transform bases, and entropy models. In future, we can exploit this flexibility to develop probability models better suited to sparse residuals and investigate alignment in other transform-based compression settings, such as graph signals and 3D Gaussian Splatting (3DGS) attributes.

AI USE STATEMENT

We used generative AI tools to assist with language editing, literature organization, figure and table preparation, and code implementation and debugging. The authors take responsibility for the manuscript, code, and reported results.

REPRODUCIBILITY STATEMENT

To support reproducibility, we describe the model architecture, coding procedure, and training settings in the main text. Our experiments use publicly available datasets, with data splits, preprocessing, and evaluation protocols specified in Section 5.1. We state the assumptions underlying our theoretical analysis and provide the corresponding derivations. The appendix supplements these descriptions with transform implementation details, analysis of the learned coefficient correction, and dataset-average rate–distortion results. We will publicly release our code, trained model weights, and data-preparation scripts to enable independent verification and further research.

REFERENCES

Johannes Ballé, Philip A. Chou, David Minnen, Saurabh Singh, Nick Johnston, Eirikur Agustsson, Sung Jin Hwang, and George Toderici. Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):339–353, 2021.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL https://arxiv.org/abs/1308.3432.

Gisle Bjøntegaard. Calculation of average PSNR differences between RD-curves. Technical Report VCEG-M33, ITU-T Video Coding Experts Group (VCEG), April 2001.

Yueru Chen, Yiting Shao, Jing Wang, Ge Li, and C-C Jay Kuo. Point cloud attribute compression via successive subspace graph transform. In *IEEE International Conference on Visual Communications and Image Processing*, pp. 66–69, 2020.

Yueru Chen, Wei Zhang, Dingquan Li, Jing Wang, and Ge Li. Hierarchical attention networks for lossless point cloud attribute compression. In *Data Compression Conference*, pp. 362–362, 2025.

Robert A. Cohen, Dong Tian, and Anthony Vetro. Attribute compression for sparse point clouds using graph transforms. In *IEEE International Conference on Image Processing*, pp. 1374–1378, 2016.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839, 2017.

Ricardo L. De Queiroz and Philip A. Chou. Compression of 3D point clouds using a region-adaptive hierarchical transform. *IEEE Transactions on Image Processing*, 25(8):3947–3956, 2016.

Tam Thuc Do, Philip A. Chou, and Gene Cheung. Learned nonlinear predictor for critically sampled 3D point cloud attribute compression. In *2024 IEEE International Conference on Image Processing (ICIP)*, pp. 3299–3305, 2024. doi: 10.1109/ICIP51287.2024.10647580.

Guangchi Fang, Qingyong Hu, Hanyun Wang, Yiling Xu, and Yulan Guo. 3DAC: Learning attribute compression for point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14819–14828, 2022.

Chunyang Fu, Tai Qin, Shiqi Wang, and Zhu Li. DeepRAHT: Learning predictive RAHT for point cloud attribute compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 14738–14746, 2026.

Wei Gao, Liang Xie, Songlin Fan, Ge Li, Shan Liu, and Wen Gao. Deep learning-based point cloud compression: An in-depth survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(11):10731–10752, 2025.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

Vivek K. Goyal. Theoretical foundations of transform coding. *IEEE Signal Processing Magazine*, 18(5):9–21, 2001.

Danillo Graziosi, Ohji Nakagami, Shinroku Kuma, Alexandre Zaghetto, Teruhiko Suzuki, and Ali Tabatabai. An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC). *APSIPA Transactions on Signal and Information Processing*, 9:e13, 2020.

Zixi Guo, Yun Zhang, Linwei Zhu, Hanli Wang, and Gangyi Jiang. TSC-PCAC: Voxel transformer and sparse convolution-based point cloud attribute compression for 3D broadcasting. *IEEE Transactions on Broadcasting*, 71(1):154–166, 2025.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs), 2016. URL `https://arxiv.org/abs/1606.08415`.

Wei Hu, Jiahao Pang, Xianming Liu, Dong Tian, Chia-Wen Lin, and Anthony Vetro. Graph signal processing for geometric data and beyond: Theory and applications. *IEEE Transactions on Multimedia*, 24:3961–3977, 2022.

Berivan Isik, Philip A. Chou, Sung Jin Hwang, Nick Johnston, and George Toderici. LVAC: Learned volumetric attribute compression for point clouds using coordinate based networks. *Frontiers in Signal Processing*, 2:1008812, 2022.

Stéphane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 3rd edition, 2009.

MPEG 3DG. Report on modifying entropy coding of attributes coefficients with dictionary removal and run-length coding. Technical Report m54265, ISO/IEC JTC1/SC29/WG11 MPEG, 2020.

MPEG 3DG. G-PCC 2nd edition codec description. Technical Report N00506, ISO/IEC JTC1/SC29/WG7 MPEG, 2023.

MPEG 3DG. Geometry based point cloud compression (G-PCC) test model version 33.0 (TMC13v33) [online]. `https://git.mpeg.expert/MPEG/3dgh/g-pcc/software/tm/mpeg-pcc-tmc13`, 2026.

Dat Thanh Nguyen and André Kaup. Lossless point cloud geometry and attribute compression using a learned conditional probability model. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(8):4337–4348, 2023.

Gaurav Pandey, James R. McBride, and Ryan M. Eustice. Ford campus vision and lidar data set. *The International Journal of Robotics Research*, 30(13):1543–1552, 2011.

Eduardo Pavez, Benjamin Girault, Antonio Ortega, and Philip A. Chou. Region adaptive graph Fourier transform for 3D point clouds. In *IEEE International Conference on Image Processing*, pp. 2726–2730, 2020.

Changhao Peng. Generalized Gaussian entropy model for point cloud attribute compression with dynamic likelihood intervals. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11779–11788, 2025.

Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, et al. Emerging MPEG standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1):133–148, 2019.

Yiting Shao, Zhaobin Zhang, Zhu Li, Kui Fan, and Ge Li. Attribute compression of 3D point clouds using Laplacian sparsity optimized graph transform. In *IEEE International Conference on Visual Communications and Image Processing*, pp. 1–4, 2017.

Xihua Sheng, Li Li, Dong Liu, Zhiwei Xiong, Zhu Li, and Feng Wu. Deep-PCAC: An end-to-end deep lossy compression framework for point cloud attributes. *IEEE Transactions on Multimedia*, 24:2617–2632, 2022.

Jonathan Taquet and S. Lasserre. Report on dyadic RAHT. Technical Report m54266, ISO/IEC JTC1/SC29/WG11 MPEG, 2020.

Jianqiang Wang and Zhan Ma. Sparse tensor-based point cloud attribute compression. In *IEEE 5th International Conference on Multimedia Information Processing and Retrieval*, pp. 59–64, 2022.

Jianqiang Wang, Dandan Ding, and Zhan Ma. Lossless point cloud attribute compression using cross-scale, cross-group, and cross-color prediction. In *Data Compression Conference*, pp. 228–237, 2023.

Jianqiang Wang, Ruixiang Xue, Jiaxin Li, Dandan Ding, Yi Lin, and Zhan Ma. A versatile point cloud compressor using universal multiscale conditional coding—part II: Attribute. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(1):252–268, 2025a.

Miaohui Wang, Runnan Huang, Wuyuan Xie, Zhan Ma, and Siwei Ma. Compression approaches for LiDAR point clouds and beyond: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 21(7):188:1–188:31, 2025b.

Lei Wei, Zhiwei Zhu, Zhecheng Wang, and Shuai Wan. Quantization parameter cascading for lossy point cloud attribute compression in G-PCC. *APSIPA Transactions on Signal and Information Processing*, 14(2), 2025.

Yiqun Xu, Wei Hu, Shanshe Wang, Xinfeng Zhang, Shiqi Wang, Siwei Ma, Zongming Guo, and Wen Gao. Predictive generalized graph Fourier transform for attribute compression of dynamic point clouds. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5):1968–1982, 2021.

Kang You, Pan Gao, and Zhan Ma. Efficient and generic point model for lossless point cloud attribute compression, 2024. URL https://arxiv.org/abs/2404.06936.

Cha Zhang, Dinei Florencio, and Charles Loop. Point cloud attribute compression with graph transform. In *IEEE International Conference on Image Processing*, pp. 2066–2070, 2014.

Junteng Zhang, Junzhe Zhang, Wenxi Ma, Dandan Ding, and Zhan Ma. Content-aware rate control for geometry-based point cloud compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(10):9550–9561, 2024.

Wenbo Zhao, Wei Gao, Dingquan Li, Jing Wang, and Guoqing Liu. LOD-PCAC: Level-of-Detail-Based Deep Lossless Point Cloud Attribute Compression. *IEEE Transactions on Image Processing*, 34:3918–3929, 2025.

A APPENDIX

This appendix provides implementation details for the TALF-based codec, including transform construction, network architecture, training, and evaluation.

A.1 TRANSFORM CONSTRUCTION

Our main instantiation uses a normalized graph-Laplacian basis (Chen et al., 2020; Pavez et al., 2020). For a block $\mathcal{B}_\nu$, its occupied children form a local graph with edge set $\mathcal{E}_\nu$. With child positions $\mathbf{r}_i$, the Gaussian adjacency weights are

$$\mathbf{A}_\nu[i,j] = \begin{cases} \exp\left(-\dfrac{\|\mathbf{r}_i - \mathbf{r}_j\|_2^2}{2\sigma_\nu^2}\right), & (i,j) \in \mathcal{E}_\nu, \\ 0, & \text{otherwise}, \end{cases}$$

where σ_ν is proportional to the cell size at the current level. Child-node positions are computed as point-count-weighted averages of their constituent point coordinates. We connect two occupied octants when their cells share a face. We set σ_ν equal to the current child-cell size. Let w_i be the number of source points represented by child u_i, and define the diagonal node-weight matrix $\mathbf{M}_\nu = \mathrm{diag}(w_1, \ldots, w_{m_\nu})$. With the degree matrix $\mathbf{D}_\nu = \mathrm{diag}(\mathbf{A}_\nu \mathbf{1})$, we construct

$$\widetilde{\mathbf{L}}_\nu = \mathbf{M}_\nu^{-1/2}(\mathbf{D}_\nu - \mathbf{A}_\nu)\mathbf{M}_\nu^{-1/2} = \mathbf{U}_\nu \mathbf{\Lambda}_\nu \mathbf{U}_\nu^\top, \qquad \mathbf{T}_\nu = \mathbf{U}_\nu^\top.$$

The eigenvectors are orthonormal, so $\mathbf{T}_\nu^\top \mathbf{T}_\nu = \mathbf{I}$. The first row produces the low-pass coefficient and is fixed to

$$\mathbf{t}_{\nu,0} = (\textstyle\sum_i w_i)^{-1/2}[\sqrt{w_1}, \ldots, \sqrt{w_{m_\nu}}],$$

while the remaining rows produce local detail coefficients and follow a deterministic sign and ordering convention.

Face connectivity does not necessarily produce a connected graph for the occupied children. When the graph has multiple connected components, its Laplacian has a multidimensional zero eigenspace. We fix the weighted constant direction as the low-pass basis and construct the remaining zero-eigenvalue vectors in its orthogonal complement; these vectors encode differences between component averages. If no edges exist, the Laplacian is zero, and the detail basis is obtained by orthogonal completion of the prescribed low-pass vector. The transform remains orthonormal and invertible. For repeated eigenvalues, we retain the deterministic eigenvector order produced by the Jacobi solver, followed by fixed reorthogonalization and sign canonicalization. The encoder and decoder derive the transform from the same decoded geometry using identical numerical conventions, requiring no additional basis information.

For node-average attributes $\mathbf{a}_\nu$ and raw encoder features $\mathbf{F}_\nu$, the point-count-weighted transform inputs used by the codec are $\mathbf{x}_\nu = \mathbf{M}_\nu^{1/2}\mathbf{a}_\nu$ and $\mathbf{H}_\nu = \mathbf{M}_\nu^{1/2}\mathbf{F}_\nu$. Applying the orthonormal transform gives

$$\mathbf{c}_\nu = \mathbf{T}_\nu \mathbf{x}_\nu, \qquad \mathbf{Z}_\nu = \mathbf{T}_\nu \mathbf{H}_\nu.$$

Reconstructed node-average attributes are recovered as $\hat{\mathbf{a}}_\nu = \mathbf{M}_\nu^{-1/2}\mathbf{T}_\nu^\top \hat{\mathbf{c}}_\nu$.

A.2 FIRST-ORDER ALIGNMENT AND RECONSTRUCTION ERROR PRESERVATION

We extend the two properties in Section 3.3 to weighted transform inputs: exact alignment of the first-order prediction term with a bounded higher-order remainder, and preservation of squared reconstruction error. The unweighted formulation in Section 3 follows by setting the weighting matrix to the identity.

Weighted transform formulation. Let $i = 1, \ldots, m_\nu$ index the elements of block ν. Associate each element with a positive weight $\alpha_{\nu,i}$ and collect them in the diagonal matrix

$$\mathbf{S}_\nu = \mathrm{diag}(\alpha_{\nu,1}, \ldots, \alpha_{\nu,m_\nu}).$$

For signal samples $\mathbf{a}_\nu$, define the weighted transform input $\mathbf{x}_\nu = \mathbf{S}_\nu \mathbf{a}_\nu$ and coefficients $\mathbf{c}_\nu = \mathbf{T}_\nu \mathbf{x}_\nu$. In the point-count-weighted construction of Section 4, $\mathbf{S}_\nu = \mathbf{M}_\nu^{1/2}$ and $\mathbf{a}_\nu$ contains node-average attributes. Weighting is applied to the input; the transform $\mathbf{T}_\nu$ remains orthonormal. Define

$$\boldsymbol{\alpha}_\nu = \mathbf{S}_\nu \mathbf{1}, \qquad \gamma_\nu = \|\boldsymbol{\alpha}_\nu\|_2.$$

Partition the analysis transform as

$$\mathbf{T}_\nu = \begin{bmatrix} \mathbf{d}_\nu^\top \\ \mathbf{T}_{\nu,\mathrm{AC}} \end{bmatrix}, \qquad \mathbf{T}_\nu^\top \mathbf{T}_\nu = \mathbf{I}.$$

The first row is the low-pass basis, and the remaining rows form the detail basis. To separate a constant signal from its detail components after weighting, the condition $\mathbf{T}_{\nu,\mathrm{AC}}\mathbf{1} = \mathbf{0}$ in Eq. (2) becomes

$$\mathbf{T}_{\nu,\mathrm{AC}}\boldsymbol{\alpha}_\nu = \mathbf{0}. \tag{11}$$

Since an orthonormal $\mathbf{T}_\nu$ has $m_\nu - 1$ independent detail rows, the nullspace of $\mathbf{T}_{\nu,\mathrm{AC}}$ is one-dimensional and is spanned by $\mathbf{d}_\nu$. Because $\boldsymbol{\alpha}_\nu \neq \mathbf{0}$, Eq. (11) holds if and only if $\mathbf{d}_\nu = \pm\boldsymbol{\alpha}_\nu/\gamma_\nu$. Fixing the positive low-pass convention gives

$$\mathbf{d}_\nu = \frac{\boldsymbol{\alpha}_\nu}{\gamma_\nu} = \frac{\mathbf{S}_\nu\mathbf{1}}{\|\mathbf{S}_\nu\mathbf{1}\|_2}. \tag{12}$$

Thus the low-pass basis follows the normalized weighted constant direction.

For a constant signal $\mathbf{a}_\nu = a\mathbf{1}$, its weighted transform input need not be elementwise constant. Instead,

$$\mathbf{x}_\nu = \mathbf{S}_\nu\mathbf{a}_\nu = a\boldsymbol{\alpha}_\nu = a\gamma_\nu\mathbf{d}_\nu.$$

It lies exactly in the low-pass direction and therefore produces

$$\mathbf{T}_\nu\mathbf{S}_\nu(a\mathbf{1}) = \begin{bmatrix} a\gamma_\nu \\ \mathbf{0} \end{bmatrix}.$$

Local first-order alignment. Stack the element features and define their weighted and aligned forms:

$$\mathbf{F}_\nu = [\mathbf{h}_{\nu,1}, \ldots, \mathbf{h}_{\nu,m_\nu}]^\top,$$
$$\mathbf{H}_\nu = \mathbf{S}_\nu\mathbf{F}_\nu, \qquad \mathbf{Z}_\nu = \mathbf{T}_\nu\mathbf{H}_\nu.$$

Assume that the signal samples admit one common nonlinear function f_ν:

$$a_{\nu,i} = f_\nu(\mathbf{h}_{\nu,i}) + \epsilon_{\nu,i}.$$

Here $\epsilon_{\nu,i}$ is the residual not explained by f_ν. The function may vary across blocks and is twice continuously differentiable in a neighborhood of the convex hull of the block's features, as in Section 3.3. Use the weighted feature center

$$\bar{\mathbf{h}}_\nu = \frac{1}{\gamma_\nu^2}\sum_{i=1}^{m_\nu}\alpha_{\nu,i}^2\mathbf{h}_{\nu,i}, \qquad \mathbf{g}_\nu = \nabla f_\nu(\bar{\mathbf{h}}_\nu).$$

Taylor's theorem gives, for each element,

$$f_\nu(\mathbf{h}_{\nu,i}) = f_\nu(\bar{\mathbf{h}}_\nu) + \mathbf{g}_\nu^\top(\mathbf{h}_{\nu,i} - \bar{\mathbf{h}}_\nu) + \tau_{\nu,i}.$$

With $\xi_\nu = f_\nu(\bar{\mathbf{h}}_\nu) - \mathbf{g}_\nu^\top\bar{\mathbf{h}}_\nu$, stacking the elements yields

$$\mathbf{a}_\nu = \mathbf{F}_\nu\mathbf{g}_\nu + \xi_\nu\mathbf{1} + \boldsymbol{\tau}_\nu + \boldsymbol{\epsilon}_\nu.$$

Multiplication by $\mathbf{S}_\nu$ sends the common intercept to $\xi_\nu\boldsymbol{\alpha}_\nu$, which is removed exactly by every detail row through Eq. (11). Consequently,

$$\mathbf{c}_{\nu,\mathrm{AC}} = \mathbf{Z}_{\nu,\mathrm{AC}}\mathbf{g}_\nu + \mathbf{T}_{\nu,\mathrm{AC}}\mathbf{S}_\nu(\boldsymbol{\tau}_\nu + \boldsymbol{\epsilon}_\nu). \tag{13}$$

The aligned features therefore represent the first-order prediction term exactly in the detail basis. The remaining terms are the transformed Taylor remainder and unexplained residual. If $\|\nabla^2 f_\nu(\mathbf{h})\|_2 \leq \kappa_\nu$ on every line segment from $\bar{\mathbf{h}}_\nu$ to an element feature, then

$$|\tau_{\nu,i}| \leq \frac{\kappa_\nu}{2}\|\mathbf{h}_{\nu,i} - \bar{\mathbf{h}}_\nu\|_2^2.$$

Combining this bound with $\|\mathbf{T}_{\nu,\mathrm{AC}}\mathbf{v}\|_2 \leq \|\mathbf{v}\|_2$ for any vector $\mathbf{v}$ gives

$$\|\mathbf{T}_{\nu,\mathrm{AC}}\mathbf{S}_\nu\boldsymbol{\tau}_\nu\|_2 \leq \frac{\kappa_\nu}{2}\left(\sum_{i=1}^{m_\nu}\alpha_{\nu,i}^2\|\mathbf{h}_{\nu,i} - \bar{\mathbf{h}}_\nu\|_2^4\right)^{1/2}. \tag{14}$$

This bound controls the higher-order contribution, separately from $\boldsymbol{\epsilon}_\nu$. For fixed weights, this contribution is small when the function has low local curvature and the block's features have limited dispersion. Setting $\mathbf{S}_\nu = \mathbf{I}$ recovers the feature center and remainder bound in Section 3.3.

Reconstruction error preservation. Independently of the local prediction model, orthonormality preserves squared reconstruction error. For any signal reconstruction $\hat{\mathbf{a}}_\nu$, let $\hat{\mathbf{x}}_\nu = \mathbf{S}_\nu \hat{\mathbf{a}}_\nu$ and $\hat{\mathbf{c}}_\nu = \mathbf{T}_\nu \hat{\mathbf{x}}_\nu$. Then

$$\|\mathbf{S}_\nu(\mathbf{a}_\nu - \hat{\mathbf{a}}_\nu)\|_2^2 = \|\mathbf{x}_\nu - \hat{\mathbf{x}}_\nu\|_2^2 = \|\mathbf{c}_\nu - \hat{\mathbf{c}}_\nu\|_2^2. \tag{15}$$

Thus coefficient-domain squared error equals weighted signal-domain squared error, reducing to ordinary squared error when $\mathbf{S}_\nu = \mathbf{I}$. For point-count weights, this is $\sum_i w_i(a_{\nu,i} - \hat{a}_{\nu,i})^2$, the point-count-weighted error of the node-average attributes. Its aggregation across the hierarchy is determined by the codec. This identity supports the coefficient-domain distortion term in Section 3.3 without requiring the local nonlinear approximation.

A.3 Network Architecture and Training

Each reference is represented by relative coordinates, reconstructed attributes, and its reconstructed prediction residual. For every target, the relative coordinates of its first 32 references are divided by their maximum absolute coordinate component. Reconstructed attributes are centered by the preliminary prediction, and both the centered attributes and reconstructed residuals are divided by 255. This gives five input channels for reflectance and nine for color. Padded references are masked during attention. The input projection maps these features to width 128. Five masked attention blocks are followed by masked sum pooling and a $128 \to 512$ adapter. The resulting child features are weighted and transformed following Section 4.1.

A shared $512 \to 256$ MLP processes each transform-aligned feature, followed by lightweight heads that predict the coefficient correction and residual scale. Only features associated with local detail coefficients are passed to these heads. The reflectance heads are linear maps from 256 to one. For color, one linear head jointly predicts the three channel corrections. The scale heads operate causally: the Cb scale head is evaluated first, followed by the Cr scale head and then the Y scale head. For each decoded residual index, we preserve its sign, apply $\log(1 + |q|)$ to its magnitude, and map the resulting scalar to a 32-dimensional embedding using a linear layer followed by ReLU. Consequently, the Cb, Cr, and Y scale heads receive 256, 288, and 320 inputs, respectively; each uses a 256-unit hidden layer and a scalar softplus output.

We train a separate model for each dataset at a fixed reference QS using the coefficient-domain rate–distortion objective. The context encoder, explicit coefficient predictor, and residual entropy model are optimized jointly. At most two million eligible blocks are sampled per epoch. A held-out subset of each training split is used for validation, hyperparameter selection, and checkpoint selection; the test sets are used only for final evaluation. The rate weights, $\beta = 1.5$ for reflectance and $\beta = 18.0$ for color, are selected empirically on the corresponding validation sets and then fixed across all experiments. The learned path is enabled after skipping the three coarsest octree levels, provided that the target has more than 16 valid references; all other coefficients use the conventional path. During evaluation, the same model is used across QS values through the conversion described in Section 4.3.

A.4 Coding and Evaluation Details

The forward pass uses hard residual quantization, and coefficient reconstruction follows Eq. (9). The quantizer uses a deadzone offset of $1/3$. The rate branch uses $q_{\mathrm{rate}} = r/\Delta + \mathrm{sg}(q - r/\Delta)$, where $r = c - \mu^{\mathrm{p}}$, q is the hard index, and sg denotes stop gradient. Thus, the forward value is the hard index while its backward derivative with respect to the continuous residual index is one. In the distortion branch, the dequantized hard residual is detached and added to the predicted coefficient; gradients therefore pass through the explicit prediction while the selected hard symbol remains fixed.

During encoding and decoding, coding proceeds from coarse to fine over the octree levels. The root low-pass coefficient and all blocks at levels containing at most 128 transform blocks are processed sequentially using the conventional run-length arithmetic backend. For each remaining level, the blocks are divided in coding order into 256 contiguous sections for reflectance and 64 for color. Processing is interleaved across sections: batching round t collects the t-th block from every nonempty section, so up to 256 reflectance blocks or 64 color blocks are evaluated in parallel. After each coding step, the reconstructed attributes are committed before the reference sets for the next step are formed. The encoder and decoder replay the same level-wise coding schedule to maintain identical causal contexts.

Table 4: Network and training configuration used for the reported models.

Configuration	Reflectance	Color
Maximum references per node	32	32
Attention blocks	5	5
Attention width	128	128
Output feature dimension	512	512
Input feature dimension	5	9
Shared MLP dimensions	$512 \to 256$	$512 \to 256$
Prediction head	$256 \to 1$	$256 \to 3$
Scale head	$256 \to 1$	$256/288/320 \to 256 \to 1$
Activations	ReLU; softplus scale	ReLU; softplus scale
Optimizer	Adam	Adam
Initial learning rate	10^{-4}	10^{-4}
Learning-rate schedule	Constant	Constant
Training epochs	40	40
Training batch size	128 (Ford), 128 (KITTI)	128
Reference QS	4	4
Rate weight β	1.5	18.0
Max. blocks per epoch	2×10^6	2×10^6
Checkpoint selection	Minimum validation objective	Minimum validation objective

Evaluation uses the data splits and preprocessing described in Section 5.1. Dataset-level bitrate and MSE are averaged across test sequences, or scans for ScanNet, and PSNR is computed from the averaged MSE. Color MSE combines the Y, Cb, and Cr channel MSEs in a 6:1:1 ratio. BD-Rate is calculated over the common quality interval using the complete operating-point sets. Runtime is measured at QS 16 for the learned-entropy variant.

A.5 Analysis of Explicit Coefficient Correction

We analyze how the integer and fractional components of the learned coefficient correction affect residual coding and reconstruction. We decompose the correction in quantization-index units as $\mu^q = \mu^q_n + \mu^q_f$, where $\mu^q_n = \mathrm{round}(\mu^q)$ is aligned with the integer quantization grid and $\mu^q_f = \mu^q - \mu^q_n$ is the remaining fractional component. The corresponding coefficient-domain components are $\mu_n = \Delta \cdot \mu^q_n$ and $\mu_f = \Delta \cdot \mu^q_f$, giving $\mu^{\mathrm{corr}} = \Delta \cdot \mu^q = \mu_n + \mu_f$.

For a translation-invariant rounding quantizer, the integer component can be absorbed exactly into the residual index. It therefore changes the coded-symbol representation without changing reconstruction, providing an entropy-oriented residual recentering mechanism. The fractional component cannot be absorbed into an integer symbol; it adjusts the reconstruction within a quantization step and may also move the residual across a quantization boundary. The actual codec uses a deadzone offset of $1/3$, for which this separation is approximate rather than exact.

The results nevertheless exhibit the expected division of roles. Integer correction concentrates the coded symbols closer to zero while only slightly changing coefficient reconstruction error (Table 5(b) and (c)), indicating that its primary effect is residual recentering. Adding the fractional component further increases the zero-symbol ratio and reduces reconstruction error across the evaluated QS settings. Although changes to the quantized symbols become less frequent at coarser QS (Table 5(a)), reconstruction still improves, consistent with fractional correction refining reconstructed values without requiring a symbol change. These results support the two practical roles of explicit prediction: residual recentering for improved symbol concentration and sub-step reconstruction refinement for quantization-aware distortion reduction.

These local analyses complement the rate–distortion results in Section 5.4.

Table 5: Mechanistic analysis of the explicit coefficient correction on Ford. The integer-only and full corrections are compared with the no-correction path using the codec's actual quantizer. Probabilities, zero-symbol ratios, and relative changes are reported in %.

(a) Learned correction decomposition

QS	$\Pr(\mu_n^q \neq 0)$	$\mathbb{E}[\lvert\mu_n^q\rvert]$	$\Pr(q_\mu \neq q_n)$
4	50.92	1.460	19.21
8	34.93	0.678	14.93
16	19.76	0.291	11.14
32	8.00	0.101	6.08
64	2.28	0.027	2.48

(b) Resulting symbol statistics

	Zero-symbol ratio			Relative increase	
QS	No μ	Integer μ_n^q	Full μ	Full vs. No μ	Full vs. Integer μ_n^q
4	34.11	37.24	38.04	11.53	2.16
8	51.85	55.59	56.73	9.41	2.05
16	68.16	71.14	72.75	6.74	2.26
32	84.46	85.99	87.38	3.45	1.61
64	94.30	94.67	95.38	1.15	0.75

	Mean absolute symbol value			Relative reduction	
QS	No μ	Integer μ_n^q	Full μ	Full vs. No μ	Full vs. Integer μ_n^q
4	2.885	2.414	2.398	16.86	0.66
8	1.323	1.105	1.086	17.91	1.69
16	0.595	0.504	0.483	18.90	4.24
32	0.226	0.197	0.181	20.23	8.15
64	0.072	0.066	0.058	19.73	11.85

(c) Coefficient reconstruction MSE

	Coefficient MSE			MSE reduction	
QS	No μ	Integer μ_n^q	Full μ	Full vs. No μ	Full vs. Integer μ_n^q
4	1.46	1.46	1.45	0.39	0.32
8	6.33	6.30	6.19	2.20	1.73
16	20.31	20.23	19.60	3.49	3.14
32	64.98	64.76	61.44	5.45	5.14
64	163.23	162.84	152.35	6.66	6.44

Let $\mathcal{Q}_\Delta(r) = \mathcal{Q}(r/\Delta)$ denote the quantized index at step Δ, and let $r_0 = c - \mu^{\mathrm{pre}}$ be the coefficient residual after the preliminary prediction. The symbols without the learned correction, with its integer component, and with the full correction are respectively

$$q_0 = \mathcal{Q}_\Delta(r_0), \qquad q_n = \mathcal{Q}_\Delta(r_0 - \mu_n), \qquad q_\mu = \mathcal{Q}_\Delta(r_0 - \mu_n - \mu_f).$$

In (a), $\Pr(\mu_n^q \neq 0)$ is the probability of a nonzero integer correction, and $\mathbb{E}[|\mu_n^q|]$ is its mean absolute value in quantization-index units. $\Pr(q_\mu \neq q_n)$ is the probability that adding the fractional correction changes the quantized symbol. In (b), the zero-symbol ratio is $\Pr(q = 0)$, and the mean absolute symbol value is $\mathbb{E}[|q|]$, evaluated for q_0, q_n, and q_μ. Panel (c) reports $\mathbb{E}[(c - \hat{c})^2]$ for the three corresponding reconstructions, where $\hat{c}_0 = \mu^{\mathrm{pre}} + \Delta \cdot q_0$, $\hat{c}_n = \mu^{\mathrm{pre}} + \mu_n + \Delta \cdot q_n$, and $\hat{c}_\mu = \mu^{\mathrm{pre}} + \mu_n + \mu_f + \Delta \cdot q_\mu$.

A.6 Complete Rate–Distortion Results

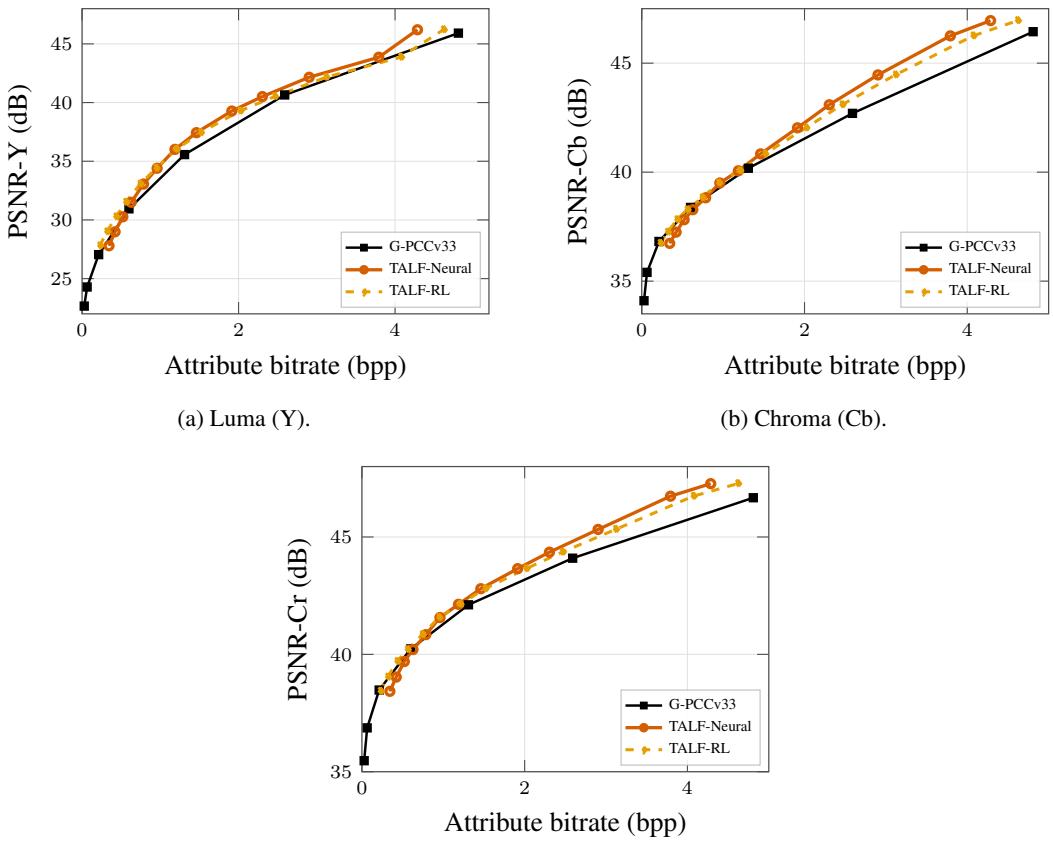
The complete dataset-average rate–distortion points for Ford, KITTI, and ScanNet are provided in Tables 6–8, with the corresponding per-channel ScanNet curves shown in Figure 4. G-PCCv33 reports the full eight-point TMC13 curve used for BD-Rate. All TALF points within each dataset are produced by a single model over the same QS range used in the main paper.

Table 6: Dataset-average rate–distortion results for Ford.

G-PCCv33			TALF		
Point	bpp	PSNR-Refl	QS	bpp	PSNR-Refl
R8	4.4898	50.17	4.00	2.9921	46.28
R7	3.5089	46.24	5.03	2.6260	43.84
R6	2.5609	41.00	6.34	2.3246	42.03
R5	1.6631	35.45	8.00	2.0329	40.16
R4	0.8915	30.23	10.06	1.8060	38.83
R3	0.3508	25.85	12.69	1.5177	36.79
R2	0.1036	22.84	16.00	1.2923	35.21
R1	0.0297	21.05	20.13	1.0602	33.44
–	–	–	25.38	0.8775	31.95
–	–	–	32.00	0.6822	30.25
–	–	–	40.25	0.5426	28.90
–	–	–	50.75	0.4165	27.55
–	–	–	64.00	0.3166	26.29

Table 7: Dataset-average rate–distortion results for KITTI.

G-PCCv33			TALF		
Point	bpp	PSNR-Refl	QS	bpp	PSNR-Refl
R8	3.6923	50.43	4.00	2.3908	46.44
R7	2.7439	46.44	5.03	2.0212	43.91
R6	1.8133	41.06	6.34	1.7255	42.07
R5	0.8855	35.56	8.00	1.4498	40.30
R4	0.2518	31.28	10.06	1.2438	38.97
R3	0.0486	29.11	12.69	0.9629	37.01
R2	0.0113	27.97	16.00	0.7651	35.57
R1	0.0043	27.31	20.13	0.5682	34.05
–	–	–	25.38	0.4235	32.88
–	–	–	32.00	0.2702	31.62
–	–	–	40.25	0.1851	30.79
–	–	–	50.75	0.1267	30.04
–	–	–	64.00	0.0945	29.45



(a) Luma (Y).

(b) Chroma (Cb).

(c) Chroma (Cr).

Figure 4: Rate–distortion curves for ScanNet, measured by (a) PSNR-Y, (b) PSNR-Cb, and (c) PSNR-Cr (dB). Bitrate is the total color-attribute bitrate in bits per point (bpp).

Table 8: Dataset-average rate–distortion results for ScanNet.

	G-PCCv33					TALF						
Point	bpp	Y	Cb	Cr	YCbCr	QS	Neural bpp	RL bpp	Y	Cb	Cr	YCbCr
R8	7.6590	49.98	49.98	49.73	49.95	4.00	4.2864	4.6183	46.20	46.95	47.27	46.41
R7	4.8095	45.92	46.39	46.62	46.06	5.03	3.7922	4.0720	43.86	46.25	46.74	44.39
R6	2.5896	40.63	42.42	43.99	41.14	6.34	2.9026	3.1165	42.16	44.46	45.32	42.69
R5	1.3095	35.49	39.82	41.97	36.33	8.00	2.3043	2.4613	40.51	43.09	44.35	41.11
R4	0.5993	30.82	38.06	40.03	31.85	10.06	1.9140	2.0176	39.28	42.03	43.65	39.92
R3	0.2118	26.84	36.48	38.21	27.96	12.69	1.4612	1.5137	37.43	40.84	42.80	38.17
R2	0.0646	24.04	35.05	36.60	25.19	16.00	1.1872	1.2057	36.01	40.06	42.13	36.82
R1	0.0277	22.42	33.71	35.16	23.58	20.13	0.9591	0.9463	34.40	39.50	41.57	35.30
–	–	–	–	–	–	25.38	0.7849	0.7428	33.06	38.83	40.84	34.00
–	–	–	–	–	–	32.00	0.6275	0.5581	31.50	38.28	40.20	32.51
–	–	–	–	–	–	40.25	0.5208	0.4323	30.27	37.82	39.69	31.31
–	–	–	–	–	–	50.75	0.4217	0.3165	28.99	37.25	39.03	30.06
–	–	–	–	–	–	64.00	0.3435	0.2267	27.80	36.73	38.43	28.89